LZT 108 3338

Mobile Phone R 320 Design Guidelines for WAP Services

ERICSSON

Ericsson is the leading provider in the new telecoms world, with communications solutions that combine telecom and datacom technologies with freedom of mobility for the user. With more than 100,000 employees in 140 countries, Ericsson simplifies communications for its customers - network operators, service providers, enterprises and consumers - the world over.

WAP™ and WAP Forum™ are registered trademarks of the WAP Forum, LTD., a registered company in the United Kingdom.

Second edition (January 2000)

This document is published by **Ericsson Mobile Communications AB**, without any warranty. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by **Ericsson Mobile Communications AB** at any time and without notice. Such changes will, however, be incorporated into new editions of this document.

All rights reserved.

© Ericsson Mobile Communications AB, 2000

Contents

PREFACE 2

Who Should Read this Guide? 2

Overview of Contents 2

Related Publications 2

Ericsson Publications 2

Other Publications 3

Terminology 3

Abbreviations 3

Definitions 3

GENERAL USABILITY ASPECTS 5

USER INTERFACE ISSUES 6

The Browser Display 6

The Keypad 7

Navigating 8

DESIGN COMPONENTS 9

Using Text 9

Font Size and Font Style 9

Line Spacing and Line Breaks 10

Paragraphs 11

The Header 12

Using Select Lists 12

Single Choice Lists 12

Multiple Choice Lists 14

Using Input Fields 16

Using Images 17

Using Hyperlinks 19

Using Images as Hyperlinks 20

Using Tables 21

Using Option Lists 23

Miscellaneous Functions 24

THE PIZZA PENGUIN EXAMPLE 25

PREFACE

This document describes how to design WAP services for the Ericsson Mobile Phone R320.

Who Should Read this Guide?

This document is intended for developers of WAP services who want a deeper insight into designing applications for the Ericsson R320 WAP browser.

It is assumed that the reader is familiar with the WAP concept and WML.

Overview of Contents

The chapters in this document are organized as follows:

General Usability Aspects

This chapter gives an overview of issues to be taken into account when developing WAP services for Ericsson R320.

User Interface Issues

This chapter gives a description of the physical and logical input/output devices of the R320 and the way these are used together with WAP services.

Design Components

This chapter contains information about graphical user interface components. The chapter also contains several examples of WML code as well as figures showing how these are presented in the browser.

The Pizza Penguin Example

This chapter contains a complete WAP application in WML code.

Related Publications

Ericsson Publications

[1] *Ericsson WAP Info*
http://www.ericsson.com/WAP/

[2] *WAP White Paper*
February 1999
AU-System AB
http://www.ericsson.com/WAP/

Other Publications

[3] *Wireless Application Protocol Architecture Specification*
Version 30-Apr-1998
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical.htm

[4] *WAP Wireless Markup Language Specification, Version 1.1*
Version 16-JUN-1999
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical.htm

Terminology

Abbreviations

GIF	Graphics Interchange Format
HTTP	Hypertext Transfer Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP™	Wireless Application Protocol
WBMP	WAP Bitmap
WML	Wireless Markup Language

Definitions

Bookmark	A named reference to a URL. Enables the user to keep track of favorite pages and then return to them, easily, later on.
Card	A single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc.
Card Contents	Subject data stored or generated at an origin server. The card contents display text, images, hyperlinks and other controls supported by WML as a response to a user request. When a user performs an action in the card contents, the WAP Browser responds according to the WML specification.
Card Title	An attribute in the WML Card element specifying advisory information about the card.
Current Card	The card whose contents is currently displayed in the browser.
Deck	A collection of WML cards.
Image	WBMP or GIF image contained in a Card.
newcontext	An attribute in the Card element indicating that the current browser context should be re-initialized upon entry to this card. The re-initialization includes removal of all previously defined variables and a reset of the navigation history.
WAP Application	A collection of WML cards, with the newcontext attribute set in the entry card.

WAP profile	A group of network and browser settings for different WAP usage scenarios. Five different WAP profiles can be utilized in a R320. See the R320 manual for details.
WAP service	A WML application residing on a web site.
WAP Services	Name of the R320 WAP browser.

GENERAL USABILITY ASPECTS

When it comes to designing any kind of electronic service, simplicity is the key to gaining acceptance and appreciation from users. Simplicity is even more important in a mobile environment since the user often needs to concentrate on other things as well as using the mobile device.

Keep information and text short and meaningful. Since overview is crucial, keep lists and menus fairly short. Do not force the user to scroll through more than 4-5 displays (corresponds to 16-20 lines). Try to avoid long words and abbreviations because the screen resources are limited. Use short and meaningful words instead of hard to understand abbreviations that might be misinterpreted.

Remember that there is a size limit to what can, in practice, be sent to the terminal in one deck. Keep the size of the decks at less than 3.000 bytes (images not included). The size of an image should not exceed 1.500 bytes. Use images to supplement text rather than replace it. Depending on service, it could be wise to stay well below the above-mentioned limits from a usability perspective. We recommend that user tests of all services be performed in a live environment.

It is better to use a few than many levels in menu hierarchies since:

1. The user can easily get lost in complex and deep menu systems.
2. Depending on the bearer and the structure of the WML code, it could be slow to move down through a multi-level menu system.

Use card titles since these are used as headers, which describe the contents of a card. Card titles should say something about the context the user is currently in. Card titles are also a navigational aid for users, helping them to remember where they are in an application.

Use links to access more information. Links are a powerful tool for linking different information items and preventing long pages of text.

Tip: *An R320 browser can be identified at the server side by the contents of the HTTP/1.1 header User-Agent request field. At the server side, knowledge of the requesting browser's identity can then be used in order to start WAP services that are tailor-made for this particular browser.*

The above-mentioned identity of the R320 browser supporting WAP 1.1 is:
"`EricssonR320/R1A`*"*

Unfortunately no standardized method exist in order to pull out this information. The methods are different depending on brand and version of the Internet server used. Please consult your Internet server documentation for details on how to make use of the information contained in the HTTP header fields.

USER INTERFACE ISSUES

The Browser Display

The Ericsson R320 features a full graphical display with a resolution of 101x65 pixels. The active screen size is about 30x23mm.

Pixel stretch is 1.24, which means that the pixels are 24% higher than they are wide. This is important to know when designing images

The browser display has two static sections: the Browser Area and the Header at the top. In addition to the static sections, dynamic overlays are used in connection with Select lists and Input fields.

Browser Area

The Browser area is where the card content is presented to the user. The size of this area is 4 lines, which corresponds to 101x52 pixels.

If a card content is too extensive to be visible in full, only the beginning of the card will be visible when first displayed. By use of the navigation keys the remainder of the card content can be viewed.

Text in the Browser area is wrapped by default.

Graphical components, text, and images are shown in the same order as they appear in the WML code.

Header

In the Header, the uppermost line of the display, the identity of the Current Card is presented. If the card has a defined title this is shown, otherwise the first 15 characters of the URL are shown (excluding the “http://”). The small bold font is used in the Header.

Tip: *Always use card titles and keep them short and meaningful.*

Overlays

Overlays are display windows that partly cover the browser area and header. Three lines at a time plus one line used as a header can be shown in an overlay window. Overlays are used in connection with Select lists and in Input fields. The user can scroll in an overlay window if it contains more lines than can be shown at a time.

The Keypad

The table below explains important key functions, used in connection with WAP Services.

Key	Normal Press	Long Press
YES	**YES** is used to select items in Single choice lists, accept an entry in Input fields, and follow hyperlinks. The **Options menu** is brought up if no field is highlighted in the Browser area.	Brings up the **Options menu** regardless of where the highlight is.
NO	Goes to the previous card. Cancels any select or input operation.	Exit to standby except when in an Input field or a select operation
C	Erases anything entered in the Input field, character by character.	Auto-repeats the erase operation.
▲ (UP)	Go to the previous line.	Autorepeat normal press.
◄ (LEFT)	Navigates left, column-by-column, in tables. Used in Multiple choice lists for checking or unchecking choices.	Autorepeat normal press.
▼ (DOWN)	Go to the next line.	Autorepeat normal press.
► (RIGHT)	Navigates right, column-by-column, in tables. Used in Multiple choice lists for checking or unchecking choices.	Autorepeat normal press.
▲ (Volume+)	Go to the previous line.	Autorepeat normal press.
▼ (Volume-)	Go to the next line.	Autorepeat normal press.
0 - 9	Used for entering numbers and characters.	
*****	Used for changing between upper and lower case.	
#	Used in Multiple choice lists for checking or unchecking choices.	

Navigating

In order to navigate to a WAP service, three methods are available. All three methods can be accessed through the **WAP Services** menu.

- Going to a homepage. There can be one homepage for each WAP profile. See the R320 User's Guide for details.
- Frequently visited locations can be bookmarked once and then reached easily using the **Bookmarks** submenu. When bookmarking, the title of the application entry card is the default name of the bookmark.
- New locations can be reached by directly entering the URL of the service, using the keypad. The **Enter Address** menu item is used in order to access this method.

Browsing through information in an application is done by scrolling using the four directional arrow keys and by activating hyperlinks.

The four directional arrow keys give a flexible and natural way to navigate between lines and links within a card. The horizontal navigation keys are used to facilitate a natural and logical navigation in tables and non-wrapped text. In all other cases vertical navigation is used.

The up and down keys scroll one line at a time.

In order to navigate backwards, the **NO** key is used. The navigation backwards can be repeated until a Card with the `newcontext` attribute is reached.

The Options Menu

The **Options menu** contains a set of very useful functions. It is a context-dependent menu, which temporarily replaces the whole browser area contents (but not the header) when the **YES** key is pressed provided that no hyperlink, input field or select list is highlighted. If the **YES** key is pressed and held down for a longer time the **Options menu** is invoked regardless of possible highlighted items.

In addition to context-dependent functions, the **Options menu** always contains useful browser functions: **Suspend**, **Reload**, **Add bookmark**, **Send as link**, **New homepage,** and **Exit**.

Options list elements (defined by WML `do` elements) are also mapped to entries in the **Options menu** (see also Using Option Lists).

DESIGN COMPONENTS

This chapter gives an overview of the graphical user interface components and descriptions of their corresponding WML elements. These components are used when designing WAP services.

The "Look and Feel" of the components is described and guidelines are made on component usage. The WML elements and their attributes are also listed and a brief description is given.

The WML element descriptions are, supplemented with WML code as well as figures, showing the way the elements are displayed and handled in the R320 browser.

Note: *The description below does not include all possible WML elements and attributes, but all elements affecting the graphical user interface.*

Using Text

Generally, text is wrapped according to word boundaries. However, if a word is longer than half the line width and starts anywhere on the first half of the line, the word is split up and wrapped. Such a split wraps an extra character in order to make room for a hyphen, making a minimum of two characters start on the new line. If such a long word starts on the second half of the display the whole word is wrapped to the next line.

Font Size and Font Style

The font used in R320 is a proportional font. A proportional font gives a dynamic and minimized width for each character, which improves readability of the displayed texts. This generally allows more characters to be displayed per line.

The used typeface name is "Ericsson Sans Serif" and two different font sizes are used: small and normal.

Font size	Character size	Line height
Small	8 pixels	13 pixels
Normal	10 pixels	13 pixels

The small and normal fonts can be shown in either regular or bold style.

WML

The R320 browser supports emphasis elements. The `big` element results in the normal font size in bold style and the `small` element results in the small font size in regular style. The `em`, `strong` and `b` elements are all handled in the same manner i.e. the text, regardless of font size, is displayed in bold. The and `u` element show the text underlined.

The `i` element is ignored.

Tip: *Be careful when using the `u` element, because underlined text normally indicates a hyperlink.*

Note: *If a paragraph is center- or right aligned, a new emphasis element within a line will cause a line break.*

A text formatting example

```
<p align="center">
   <b><big><a href="#Second">Welcome</a></big></b><br/>
   <small>to</small><br/>
   The Mobile Way of Ordering Pizza.<br/>
   <a href="#Fifth">[Contact Us]</a>
</p>
```

Line Spacing and Line Breaks

The line spacing in the R320 browser is single line. If a line is too long to fit horizontally in the browser area, the line will by default be wrapped onto multiple lines.

WML

The beginning of a new line is defined by a `br` element.

The `br` element affects all contents in the browser view. Text, select lists, input fields, hyperlinks and tables are positioned on a new line according to the current alignment. It can also be used inside a table cell (see Using Tables below).

The br element should not be used within `a`- or `anchor` tags since line breaks within hyperlinks might reduce readability.

Tip: *Do not use `br` elements within hyperlinks, because this might impair readability.*

A line break example

The example below shows how the `br` element will affect the card content appearance.

```
<wml>
<card id="init" title="BR tag">
<p>
   This text will wrap to a new line.
   <br/><br/>
   Preceded by 2 br tags.
</p>
</card>
</wml>
```

Paragraphs

Text on a WML card comprises one or more paragraphs.

A paragraph always starts on a new line.

Text within a paragraph is by default displayed left aligned. However, it can optionally be displayed right aligned or centered.

WML

Paragraphs are defined by using the `p` element.

The `p` element also establishes the alignment attribute for a paragraph.

Attribute	Description
`align`	The `align` attribute can take the following values: *left*, *right* and *center*. Left alignment is the default alignment mode.
`mode`	The `mode` attribute specifies the line-wrap mode for the paragraph and can take the following values: *wrap*, which specifies breaking text mode and *nowrap*, which specifies non-breaking text mode. Breaking text mode is the default mode.

A paragraph example

The example below shows how the `p` element and `align` attribute affect the card content appearance.

```
<wml>
<card id="init" title="P tag">
<p>
   This text is left aligned.
</p>
<p align="center">
   This is centered.
</p>
<p align="right">
   Right aligned.
</p>
</card>
</wml>
```

The Header

See **Header** on page 6.

WML

The card identity in the Header is defined by using the `title` attribute in the `card` element.

Using Select Lists

A select list is an input component that specifies a list of options for the user to choose from. Single and multiple choice lists are supported.

Single Choice Lists

A single choice list is used for the selection of one and only one item in a list. A single choice list is accessed from the browser area through Select link. A Select link is highlighted by inverse video when in focus.

In its closed state (as shown in the browser area) the single choice list displays the currently selected item within angled brackets (see *Select Size* in the example below). If the user has not previously selected a value, the first item in a single choice list is the default value and thus displayed within the angled brackets.

If the available space after the Select link is too small (smaller than the space needed for the text “WWW”) the value including brackets is wrapped to the next line. Otherwise the value will be truncated to fit on the same line as the Select link.

In its opened state (after the user has pressed the **YES** key on a highlighted Select link) the single choice list is presented in an overlay window as a group of radio buttons, one button for each choice. If a choice text preceding a radio button is too long, it is truncated.

The user navigates to the desired item and presses **YES** in order to select and return.

After returning from the overlay window, by pressing **YES** or **NO**, the now currently selected item is displayed within angled brackets and the Select link is still highlighted.

WML

Two elements define a single choice list: `select` and `option`.

Single choice lists are specified using the `select` element with the `multiple` attribute set to *No*. Each list item is specified by an `option` element.

The `select` element

Attribute	Description
`multiple`	The `multiple` attribute can take the following values: *Yes* and *No*. The default value is *No*. To create a single choice list, this attribute is either omitted or set to *No*.
`title`	The value of the `title` attribute is used both as Select link text and as header in the overlay window

Tip: *It is recommended to include the title attribute. However, if the title attribute is left out, the choice list will be headed "Select" translated into current display language.*

The `option` element

Text in the `option` element content is used as a description of the choice.

The `title` attribute is not supported for the `option` element.

An example of a single choice list

```
<p>
   <select title="Type">
      <option>Cappriciosa</option>
      <option>Quatro Staggioni</option>
      <option>Margherita</option>
      <option>Campagnola</option>
      <option>Siciliana</option>
   </select>
</p>

<p>
   <select title="Size">
      <option>Large</option>
      <option>Medium</option>
      <option>Small</option>
   </select>
</p>
```

Multiple Choice Lists

A multiple choice list is used for the selection of zero, one or more items in a list. A multiple choice list is accessed from the browser area through a Select link.

In its closed state (as shown in the browser area) the multiple choice list displays the currently selected items, separated by commas, within angled brackets (see *Select Extras* in the example below). The text within the brackets is truncated if necessary. If no item is selected, only the brackets are displayed.

If the available space after the Select link is too small (smaller than the space needed for the text “WWW”) the text including brackets is wrapped to the next line.

In its opened state (after the user has pressed the **YES** key on a highlighted Select link) the multiple choice list is presented in an overlay window as a number of check boxes, one for each choice. If a choice text preceding the check box is too long, it is truncated.

The user navigates to the items to change and uses the left and right arrows or the **#** key in order to check and uncheck the boxes.

After returning from the overlay window, by pressing **YES** or **NO**, the now currently selected items are displayed within angled brackets and the Select link is still highlighted.

WML

Two elements define a multiple choice list: `select` and `option`.

Multiple choice lists are specified using the `select` element with the `multiple` attribute set to *Yes*. Each check box item is specified by an `option` element.

The `select` element

Attribute	Description
`multiple`	The `multiple` attribute can take the following values: *Yes* and *No*. The default value is *No*. To create a multiple choice list, this attribute is set to *Yes*.
`title`	The value of the `title` attribute is used both as Select link text and as header in the overlay window.

Tip: *It is recommended to include the title attribute. However if the title attribute is left out, the choice list will be headed "Select" translated into current display language.*

The `option` element

Text in the `option` element content is displayed to the right of the check box.

The `title` attribute is not supported for the `option` element.

An example of a multiple choice list

```
<p>
   <select title="Extras" iname="I" multiple="true">
      <option>Mushrooms</option>
      <option>Mozzarella</option>
      <option>Garlic</option>
   </select>
</p>
```

Using Input Fields

Input fields are the only components enabling a user to enter text.

Input fields are accessed from the browser area through an Input link leading to an overlay window in which a new text can be entered or an already entered text can be edited. An Input link is highlighted by inverse video when in focus. After returning from the text input, by pressing **YES** or **NO,** the now current value of the text field is displayed within angled brackets.

If the available space after the Input link is too small (smaller than the space needed for the text "WWW") the text including brackets is wrapped to the next line. Otherwise the text value is truncated to fit on the same line as the Input link. The Input link is still highlighted.

The text in an input field is displayed using the Normal bold font in the overlay window.

WML

Input fields are defined by using the `input` element.

The following attributes are supported:

Attribute	Description
`type`	The `type` attribute can take the following values: *text* and *password*. If `type` is set to *password*, asterisks will replace the characters.
`value`	If no pre-load value is defined for the input object, the value of the `value` attribute is used as a default text in the input field.

`format`	Specifies an input mask for user input entries. The string consists of mask control characters and static text that is displayed in the input area. The format control characters specify the data format expected to be entered by the user. The format codes are: **`A`** Upper case alphabetic or punctuation, **`a`** Lower case alphabetic or punctuation, **`N`** Digits, **`X`** Upper case alphabetic character, **`x`** Lower case alphabetic character, **`M`** Any character, **`m`** Any character, **`*f`** Any number of characters of type f, **`nf`** n number of characters of type f, **`\c`** Display the special character c, in the entry field.
`maxlength`	Maximum number of characters a user can enter.
`title`	The value of the `title` attribute is used both as Input link text and as header in the overlay window.

Tip: *It is recommended to include the title attribute. However, if the title attribute is left out, the input field will be headed "Input" translated into current display language.*

An input example

```
<p>
   <input type="text" title="Enter name:" name="name"
   format="*A"/>
   <input type="text" title="Enter address:" name="address"
   format="*M"/>
</p>
```

Using Images

The R320 browser supports images in WAP bitmap (WBMP) and GIF formats. The size of an image in WAP bitmap format must not exceed 1500 bytes. For images in GIF format the corresponding size restriction is that image width multiplied by image height (both values in pixels) must not exceed 12000.

An image is always loaded after the WML code. During the load time, the browser displays a rectangular placeholder with an optional text. The size of the placeholder depends on the length of the optional text. See the figure below.

However, the R320 user can choose not to load images at all. In this case, placeholders become permanent.

A margin of 4 pixels at the top and 4 pixels at the bottom is added to the image space. An image is displayed on separate lines and uses the minimum number of complete lines (13 pixels) needed to fulfil the height of an image.

Images are shown centred horizontally and centred vertically.

There is no limitation to the height of an image. An image that is higher than the display can be viewed by scrolling up and down one line (13 pixels) at a time. However it is recommended to use images not higher that the display height. The maximum image height to be fitted on one screen without need for scrolling is 4 * 13 - 4 pixels.

If an image is wider than the browser area, it is centered and truncated at both ends.

WML

Images are defined by the `img` element.

The R320 supports the following attributes:

Attribute	Description
`alt`	An alternative textual representation for the image used in the placeholder.
`src`	The source (URI) of the image.
`height`	This attribute indicates the vertical size of an image.
`width`	This attribute indicates the horizontal size of an image.

The `height` and `width` attributes are only supported when specified as absolute values.

An image example

```
<p align="center">
   <img alt="baker"src="baker.gif" width="40" height="30"/>
   The oven is turned on, and your pizza will be delivered just
in time.
</p>
```

Using Hyperlinks

The concept of hyperlinks is well known from the Internet environment. All hyperlinks in the R320 are underlined to mark that they are hyperlinks. When a hyperlink is in focus it is highlighted by inverse video.

Some examples of hyperlink usage:

- Navigate within a deck.
- Navigate to a new deck.
- Start composing an SMS.
  The Send New function is started and the e-mail address from the mailto URL is copied into the message area.
- Download a new melody.
  The user is prompted to play and save the new melody.
- Download an electronic business card.
  The user is prompted to save the business card.
- Initiate a voice call.

A hyperlink may be located anywhere in the text flow. Text in hyperlinks is displayed in the same font as the text preceding the link, but the text is underlined. Selecting the link invokes the action associated to the link.

WML

Hyperlinks are specified using the `a` element.

Below are three examples of useful functions in the R320 that can be activated through hyperlinks.

Download a new melody

```
<a href="Bamse.mel"> Bamse </a>
```

The file must have a `.mel` extension or the content type must be set to `text/x-vmel` at the server.

See Melody Composer found at Ericsson Mobile Internet for details on how to compose and store melodies.

Download an electronic business card

```
<a href="LarsMagnusEricsson.vcf"> Download Lars-Magnus Ericssons
Business Card </a>
```

The file must have a `.vcf` extension or the content type must be set to `text/x-vcard` at the server.

Start compose an SMS

```
<a href="mailto:pizza@penguin.nu">Send an SMS to Pizza
Penguin</a>
```

A hyperlink example

Three different links are shown in the example: a mailto link, a voice call link and an interdeck link.

```
<card id="Fifth" title="Contact Us">
<p align="center">
   <br/><a href="mailto:pizza@penguin.nu?&#38;subject=Pizza">
   Send a mail to Pizza Penguin</a>
   <br/><br/><a href="wtai://wp/mc;+464690510">
   Call Pizza Penguin</a><br/><br/>
</p>
<p align="center">
   <a href="#First">[Go to Start]</a>
</p>
</card>
```

Using Images as Hyperlinks

An image can also serve as a hyperlink. Images that are used as hyperlinks are marked with a 2 pixels wide frame around the image. The space for the frame is taken from the image space. This implies that images that are used as hyperlinks should not be too small.

WML

An image hyperlink example

```
<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu">
   <img alt="mail"src="mail.gif" width="40" height="40" /> </a>
   <a href="wtai://wp/mc;+464690510">
   <img alt="phone"src="phone.gif"  width="40" height="40"/>
   </a>
</p>
```

Using Tables

Due to its large display and four-directional navigation possibilities, the R320 browser has good support for tables. Tables are used to make information easier to grasp and analyze.

The R320 browser displays a table in columns and rows with solid borders. A table is located on a new line. Empty table cells and empty table rows are significant and are included in the presentation of the table.

When a table is in focus it can be panned, column-by-column, using the horizontal navigation keys in order to view all columns.

The minimum column width is 20 pixels. The maximum number of rows is 80 and the maximum number of columns is 20. Rows and columns exceeding these maximum values are discarded.

Text in a table cell can be left aligned, centre aligned or right aligned. All text-formatting rules apply to text within a table. The browser will handle a multiple-lines cell, which may result from wrapping of cell contents.

Tables are presented column by column on the screen. Each column width is calculated from the widest single cell up to a maximum of 80% of the display width, i.e. 81 pixels. Text content which is wider than this is wrapped to fit the display. This wrapping tries to put as much text as possible in a row and therefore wraps all characters, plus one to fit a hyphen, to the next row.

Tables can also be used to format card contents that include images in an appealing way. If for example you would like to have an image followed by a text on the same row the image is put in the first column and the text in the second.

A margin of 4 pixels at the top and 4 pixels at the bottom is added to the image space. If an image is wider than 80% of the cell width, the part of the image that is larger than 80% is truncated. There is no principal height limitation to an image.

In addition to text and images, tables can also contain hyperlinks and images used as hyperlinks.

WML

The `table` element is used together with `tr` and `td` to create a set of aligned columns of text and images.

The `table` element

Attribute	Description
`align`	Alignment of cell content is fully supported. The `align` attribute can take the following values: *L* (left), *R* (right), and *C* (centre). Left alignment is the default alignment mode.
`columns`	Specifies the number of columns for the table.

The `tr` and `td` elements

The `tr` element is used as a container to hold a single table row.

The `td` element is used as a container to hold a single table cell data.

The Polar Weather `table` example

```
<card id="First" title="Polar Weather" newcontext="true">

<p>
   <table align="ccccc" columns="5">
   <tr>
      <td>Monday</td>
      <td>Tuesday</td>
      <td>Wednesday</td>
      <td>Thursday</td>
      <td>Friday</td>
   </tr>

   <tr>
      <td><img alt="weather"src="snow.gif" width="17"
      height="17" /></td>
      <td><img alt="weather"src="snow.gif" width="17"
      height="17" /></td>
      <td><img alt="weather"src="sun.gif" width="17"
      height="17" /></td>
      <td><img alt="weather"src="sun.gif" width="17"
      height="17" /></td>
      <td><img alt="weather"src="sun_in_cloud.gif" width="17"
      height="17" /></td>
   </tr>

   <tr>
      <td>Snow</td>
      <td>Snow</td>
      <td>Sunny</td>
      <td>Sunny</td>
      <td>Partly Cloudy</td>
   </tr>

   <tr>
      <td>0&#176;C</td>
      <td>-1&#176;C</td>
      <td>-5&#176;C</td>
      <td>-5&#176;C</td>
      <td>-1&#176;C</td>
   </tr>
   </table>
</p>
</card>
```

Using Option Lists

Option lists are components that start actions or change states.

All Option list elements are mapped to entries in the **Options menu**. These dynamic menu items precede the static functions in the **Options menu**.

Note: *Use hyperlinks instead of Option list elements since hyperlinks are more accessible in the user interface.*

WML

Option list elements are defined by using the WML `do` element.

Selecting the associated menu item in the **Options menu** invokes the action defined in the `do` element.

The R320 supports the following attributes:

Attribute	Description
`label`	The value of the `label` attribute is used as text in the Options menu. The text is not truncated.
`type`	If no `label` is specified, then the value of the `type` attribute is used in the Options menu. The `type` attribute can take the following values: *accept, prev, help, reset, options, delete,* and *unknown.*

Tip: *The Option list elements will be ordered by their type attributes in the options menu. An Option list element with the `type` accept will be placed first, followed by an Option list element with the `type` prev. Following these, all other Option list elements are placed in the order they appear in the card.*

Note: *If two do elements have identical `type` values, `name` attributes must be specified in order to create unique bindings.*

An Option List example

```
<p align = "center">
   <do type="accept" label="What's On">
   <go href="#PizzaInfo"/>
   </do>
   .
   .
</p>
```

Miscellaneous Functions

Start WAP Services from an SMS

If an SMS message contains the URL of a WAP service, the user can select the URL, press **YES** and the URL will automatically be copied into the **Enter Address** function. If the user presses **YES** again R320 browser will be started and the WAP service loaded.

Send as link

The **Option menu** contains a function **Send as link** that enables the user to send the URL of the Current Card in an SMS to another phone.

THE PIZZA PENGUIN EXAMPLE

Below is a complete WML code listing from a WAP demonstration application. The examples in the previous chapters were mainly selected from this application.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card id="First" title="Pizza Penguin" newcontext="true">
<onevent type="ontimer">
<go href="#Second"/>
</onevent>
<timer value="70"/>

<p>
   <img alt="Penguin"src="mpenguin.gif" width="45" height="45"/>
</p>
</card>


<card id="nop" title="nop">
<p>
   <a href="#Second"><img alt="Penguin" src="mpenguin.gif"
   width="45" height="48"/></a>
</p>
</card>


<card id="Second" title="Select Pizza">
<p>
   <select title="Type">
      <option>Cappriciosa</option>
      <option>Quatro Staggioni</option>
      <option>Margherita</option>
      <option>Campagnola</option>
      <option>Siciliana</option>
   </select>
</p>

<p>
   <select title="Size">
      <option>Large</option>
      <option>Medium</option>
      <option>Small</option>
   </select>
</p>

<p>
   <select title="Extras" iname="I" multiple="true">
      <option>Mushrooms</option>
      <option>Mozzarella</option>
      <option>Garlic</option>
   </select>
</p>
```

```
<p align = "center">
   <do type="accept" label="What's On">
   <go href="#PizzaInfo"/>
   </do>
   <a href="#Third">Continue</a>
   <a href="#Fifth">[Contact Us]</a>
   <a href="#First">[Go to Start]</a>
</p>
</card>


<card id="Third" title="Name &amp;amp; Address">
<p>
   <input type="text" title="Enter name:" name="name"
   format="*A"/>
   <input type="text" title="Enter address:" name="address"
   format="*M"/>
</p>

<p align="center">
   <br/>
   <a href="#Sixth">Submit order</a>
   <a href="#Fifth">[Contact Us]</a>
   <a href="#First">[Go to Start]</a>
</p>
</card>


<card id="Fifth" title="Contact Us">
<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu">
   <img alt="mail"src="mail.gif" width="40" height="40" /> </a>
   <a href="wtai://wp/mc;+464690510">
   <img alt="phone"src="phone.gif"  width="40" height="40"/>
   </a>
</p>

<p align="center">
   <a href="#First">[Go to Start]</a>
</p>
</card>


<card id="Sixth" title="Oven On">
<p align="center">
   <img alt="baker"src="baker.gif" width="40" height="30"/>
   The oven is turned on, and your pizza will be delivered just
in time.
</p>

<p align="center">
   <a href="#Fifth">[Contact Us]</a>
   <a href="#First">[More Pizzas]</a>
</p>
</card>
```

```
<card id="PizzaInfo" title="What's On">

<p>
   <b>Cappriciosa </b>
   Mozzarella, tomato sauce, prosciutto, olives, artichokes
   &amp;amp; mushrooms

   <b>Quatro Staggioni </b>
   Tomato, mozzarella, ham, artichokes &amp;amp; mushrooms
   <b>Margherita </b>
   Fresh tomatoes, fresh mozzarella &amp;amp; basil

   <b>Campagnola </b>
   Tomato, mozzarella, zucchini, eggplant, peppers
   &amp;amp; mushrooms

   <b>Siciliana </b>
   Carciofini, prosciutto, pomidoro, fresh mozzarella
   &amp;amp; whustel
</p>

<p align="center">
   <a href="#Fifth">[Contact Us]</a>
    <a href="#First">[Go to Start]</a>
</p>

</card>

</wml>
```